Generic technique and the dynamics of technologies: using matroid and design theory to design techniques with systemic impact

Pascal Le Masson, Armand Hatchuel, Olga Kokshagina, Benoit Weil

ICED15 27-30 JULY - MILANO

GENERIC TECHNIQUE AND THE DYNAMICS OF TECHNOLOGIES: USING MATROID AND DESIGN THEORY TO DESIGN TECHNIQUES WITH SYSTEMIC IMPACT

Pascal Le Masson, Armand Hatchuel, Olga Kokshagina, Benoit Weil
Mines ParisTech, France

Abstract

As underlined in Arthur's book "the nature of technology" we are very knowledgeable on the design of objects, services or technical systems, but we don't know much on the dynamics of technologies. Still contemporary innovation often consists in designing techniques with systemic impact. They are pervasive– both invasive and perturbing-, they recompose the family of techniques. Can we model the impact and the design of such techniques? More specifically: how can one design generic technology, ie a single technology that provokes a complete reordering of families of techniques?
Recent advances in design theories open new possibilities to answer these questions. In this paper we use C-K design theory and a matroid-based model of the set of techniques to propose a new model (C-K/Ma) of the dynamics of techniques, accounting for the design of generic technologies. We show:
• F1: C-K/Ma offers a computational model for designing a technique with systemic impact.
• F2: C-K/Ma accounts for some phenomena associated to generic technology design.
• F3: C-K/Ma offers an efficient guide for the design of technologies with systemic impact, based on generativity and genericity criteria

1 INTRODUCTION – DESIGNING FOR SYSTEMIC IMPACT?

In his book "the nature of technology" (Arthur 2009), W. Brian Arthur explains that looking for "some common logic that would structure technology and determine its ways and progress" he couldn't find it. Is this claim exaggerated in the light of the literature in the engineering design community? Not that much: we are very knowledgeable on the design of objects, services or technical systems, relying on techniques or building blocks. But what do we know on the design of these "techniques" or building blocks? What do we know about the dynamics of techniques?
This question is all the more relevant when we consider some famous technologies and their impact: software in mechanical systems (aeronautics, mechatronics), semiconductors in a large variety of systems, additive processes in industrial processes, etc. All these technologies have a "systemic" impact, they are *pervasive* – both invasive and perturbing-, they recompose the family of techniques. Of course we know of some technologies that completely changed the technical environment of their time – "steam engine", "electricity"… One even associated each industrial revolution to a handful of such techniques. And today this process of *reorganizing* the family of techniques might be more frequent. And it is not sure that we understand it: what is the origin of such pervasive techniques? How can we model their impact - rearrangements, "creative destruction", re-ordering? Are there design strategies to design the techniques and their impact? And even more specifically: how can one design generic technology, ie a single technology that provokes a complete reordering of families of techniques?
Recent advances in design theories open new possibilities to answer these questions. In this paper we use C-K design theory and a matroid-based model of the set of techniques to propose a new model (C-K/Ma) of the dynamics of techniques, accounting for the design of generic technologies. We show:

- F1: C-K/Ma offers a computational model for designing a technique with systemic impact.
- F2: C-K/Ma accounts for some phenomena associated to generic technology design.
- F3: C-K/Ma offers an efficient guide for the design of technologies with systemic impact, based on generativity and genericity criteria.

2 RESEARCH QUESTIONS – DESIGN AND INTERDEPENDENCES

2.1 Endogenous dynamics of technologies : the design of interdependencies

In the multiple approaches on the dynamics of technologies, we can distinguish two main trends. On the one hand, many works tend to consider that the dynamics of technologies is based on combinations. Arthur's book provides a synthesis of these approaches (Arthur 2009). Technologies are building blocks that can be "*combined*". Combinations are selected – for instance by markets. The scientific study of phenomena regularly provides new building blocks. Hence the dynamics of technologies is combinatorial, and it is controlled by market and science. This kind of model is used in many evolutionary economics works (Dosi et al. 1988; Saviotti and Metcalfe 1991). This approach is "exogenous": the dynamics of technologies relies on exogenous forces – market and science.

On the other hand, several authors, particularly in engineering design have underlined that for one given set of functions, there are different ways –different combinations- to address it. In particular, Design Structure Matrices (Ulrich and Eppinger 2008), or Modularity (Baldwin and Clark 2000), or Aximatic Design (Suh 2001) lead to distinguish between technological systems, although these systems seem equivalent from a functional point of view: systems with less interdependencies (DSM), with modularity (Baldwin's Design Rules) or with independences (first axiom of Axioamtic Design) are "better" – they are said to be more robust, easier to realise, or they enable a large variety of alternatives,… In these approaches, market and science are not the only engines in the dynamics of technological system: the techniques themselves have their own, *endogenous dynamics*.

This endogenous dynamics is related to some "structures" of the technical systems. Engineering design literature has already studied how, in certain cases, specific interdependences have a *systemic impact:* enabling "modularity" or "diagonal matrices" or providing a new "common core" (Gawer 2009; Kokshagina et al. 2013a) open access to a large variety of configurations. This systemic impact can be characterized by criteria like generativity and genericity: the generativity of the design in a technical system is the variety and originality that can be designed with a given system (Le Masson and Weil 2013; Hatchuel et al. 2011)), whereas one single technique can be used in many different systems, this is the genericity of this technique (Bresnahan and Trajtenberg 1995).

The literature underlines the deep interaction between techniques interdependences and design. Still we miss a general model for *the "endogenous" dynamics of technology,* ie for *design strategies with systemic impact* (**Q_0**). More precisely:

- **Q_1**: Can we find a *model that accounts rigorously for "systemic impact",* and in particular for such notions as genericity, generativity, independence and dependence - can we *reach a computational and quantitative approach of these notions*?
- **Q_2**: How does such a model predict the emergence and possibility of *strategies for the design of techniques with systemic impact (pervasive techniques)*? In particular, how can one model the *design of generic technologies*?

2.2 Design theory and knowledge structure

How do design theories account for "systemic impact"?

1- Contribution to Q_1: design theories account largely for the impact of knowledge structures on the design of new artefacts. Historical analysis of Design Theories studied the effect of knowledge structure on the generative power in different cases (parametric design, systematic design) (Le Masson and Weil 2013). General Design Theory (Yoshikawa 1981; Reich 1995) shows how a knowledge structure with an Hausdorff measure (distinction criteria) warranties the design of any functional combination; Coupled Design Process (Braha and Reich 2003) shows that more generally, the set of functional combinations that can be reached depends on the "closure" of technologies, ie their neighbourhood of alternative technologies; in Infused Design (Shai and Reich 2004a, b), the design capacity depends on the rules in one domain and the laws linking different domains and it has been shown that innovative design comes from the existence of "holes" in the correspondence between domains (Shai et al. 2013); in C-K theory the generative power comes from "holes" in the knowledge structure (Hatchuel et al. 2013).

This review shows that *given a structure of dependences and independences in knowledge, Design Theories help to predict the impact of this structure on design capacities*, characterized by differences in the capacity to create original design (*generativity*) and in the capacity to generate simultaneously or

easily a large set of solutions (*genericity*). *However they don't offer yet a computational and quantitative approach of generativity, genericity, dependence and independence.*
2- Contribution to Q_2: how do design theories model the design of a new "technology" characterized by its systemic impact? Or: how to design a specific "knowledge structure"? Actually design theories tend to favour the design of artefacts, given a certain knowledge structure; they barely address the issue of designing a specific knowledge structure. Some insights are given in (Kokshagina et al. 2013a), based on a study of algebraic extensions with C-K theory: the authors show that an algebraic extension begins with a concept and, depending on the concept, it is possible to generate fields with specific structures – in particular a specific field size. In the example, the authors studied the design process associated to a knowledge structure. It was actually made possible by one key property of contemporary design theories: they don't depend on specific objects or domains – knowledge is a "free parameter".
Hence, even if design theories did not focus until now on the design of specific knowledge structures, design theories provide favourable frameworks to study this question. *In this paper, we will use design theories to analyse design strategies applied on knowledge structures.*

2.3 Knowledge structures characterized (only) by independences: matroids

What is the relevant knowledge structure to study design strategies with systemic impact? As seen above, we expect to *characterize this knowledge structure by dependences and interdependences*, such that generativity and genericity can be computed.
Let's take an example: following axiomatic design, the designer diagonalizes a matrix of technologies: he designs a technique to change the (interdependent) techniques into independent ones. In this case, interdependences in the knowledge base are characterized by linear algebra. We could also take as knowledge base a functional graph (with graphical relations between functions) and design a graph with less interdependences. We use design theory to change the dependence relations in a graph. Yet, in these examples, we are actually only interested by the *evolution of dependence relations* – be they based on linear algebra or graphs. Hence our analysis would be more general if applied to a knowledge structure that is only characterized by the interdependences between the elements – whatever the deep nature of the relation (graph or linear algebra). By chance mathematicians have already studied such a strange object in great detail: they call it *matroid*.
Matroid structures were introduced by Whitney, in the 1930s (Whitney 1935), to capture abstractly the essence of (linear) dependence. Whitney explains his project as follows: "let C_1, C_2,… C_n be the columns of a matrix M. Any subset of these columns is either linearly independent or linearly dependent; the subset thus falls into two classes. These classes are not arbitrary; for instance the two following theorems must hold: a) any subset of an independent set is independent; b) if N_p and N_{p+1} are independent sets of p and p+1 columns respectively then N_p together with some column of N_{p+1} forms an independent set of p+1 columns […] Let us call a system obeying a) and b) a "matroid". (p. 509) (Whitney 1935). Hence the description with matroids will be very general (and quite poor – as Whitney: "The fundamental question of completely characterizing systems which represent matrices is left unsolved") but it has *the great advantage of only characterizing the relationship between elements in two modes: independence and dependence*. And this remains valid for many types of relations– Whitney: "In place of a matrix we may equally well consider points or vectors in a Euclidean space, or polynomials, etc…" and more recently: graphs, matrices, groups, algebraic extensions… (for a pedagogical introduction to matroid, see (Neel and Neudauer 2009)).

Hence this paper aims at understanding the '*endogenous*' dynamics of technologies by applying design theory on a knowledge structure modelled with *matroid*. We expect to characterize in this model how design strategies depend on (and also change) dependences and independences in a knowledge structure and how specific design strategies increase generativity and genericity.

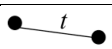
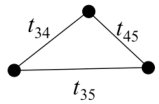
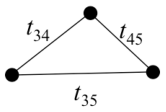
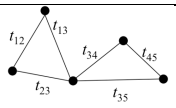
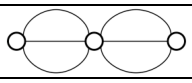
3 A C-K MODEL WITH MATROIDS

3.1 Notions of techniques, systems and independences in a matroid framework

To build a C-K model with matroid, we introduce the matroid of known techniques as the K-space in C-K. Assume E, a list of known techniques (also called technological building blocks in the paper): T_1,… T_n. We select a subset of this set of technological building blocks. If they build a working

system, we will say they are *dependent* (we could say "compatible"); if not, they are *independent*. We can easily check that these sets follow the matroid properties, hence form a matroid.
We model in matroid basic notions of the dynamics of technologies: a technique, a working system, a family of techniques, the structure of all techniques, the structure of all working systems (See table 1).

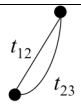
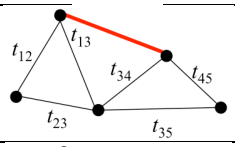
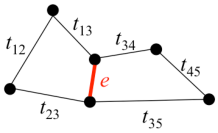
Table 1. notions in the dynamics of techniques and notions in matroid theory

Dynamics of techniques	Matroid theory	Illustration with graphic
Technique	An element in a matroid	
Working system: a system made of compatible techniques (techniques that work together)	A circuit (eg a cycle, ie a minimal circuit)	 extracted from M below
A family of techniques: a subset of techniques such that no "external" technique is compatible with the techniques in the family	A flat, extracted from a matroid	 or:  extracted from M below
The structure of techniques	The matroid of techniques	Matroid M: 
The structure of working systems	The dual of the matroid of techniques	

3.2 Designing a new matroid: a C-K design process on matroids

The design process is modeled with C-K theory with a matroid in K. It enables critical distinctions:
a) C-K/Ma accounts for the *distinction between deduction and design*. Deduction consists in proving that one circuit (working system) exists in the matroid; design consists in creating techniques added to the matroid, to design one circuit that was "unknown" in the initial matroid (a new working system).
b) C-K/Ma accounts for the distinction between designing a "stand alone" working system based on existing techniques (one "project") and designing a structured set of techniques: the first one consists in extracting one working system from the matroid of known techniques to create a separate working system (and matroid theory teaches us when and how it is systematically possible); the latter consists in designing a new technique added to the matroid of techniques.
We are now focusing on this latter case: the design of a new matroid of techniques from an existing one, by adding one technique (one edge). Matroid theory teaches us that there are only two operations to add one additional edge to a matroid: extension and co-extension (Oxley 2011) – see table 2 below. We can show (Le Masson et al. 2015) that these two operations in matroid correspond to two ways to design techniques: designing for one new working system or designing a generic technique.

Table 2. Design operations in the dynamics of technique and in matroid

Designing one working system based on existing techniques	Extracting one circuit (possible for all minors of the matroid of techniques)	
Cumulative design of working systems with new technique linking other techniques and minimizing propagations	Extension ie one dependent edge, depending on the techniques to be linked together	
Designing a generic technique, generic to several technical families	Coextension, ie one independent edge common to several connected components	

1- Matroid extension models the design of one technique to get one additional working system inside one family, while keeping all the already known techniques and their relations. It is *as little pervasive as possible*: extension only changes the families of techniques (flats) containing the new working system – all the other families are unchanged.

2- Co-extension is, in matroid, the dual operation. It is far less intuitive. It models the design of a "pervasive" (or generic) technique. We can illustrate the coextension logic, and its difference with extension, on one example (figure 1) : suppose one knows techniques to make knives and techniques to make bottle openers. By extension one can take one technique of the "family" of knives and another from the family of bottle openers and make them dependent by designing one additional technique – namely the insertion of a bottle opener in the knife handle: this is an extension. By co-extension one designs a technique to keep all the previous knifes and bottle openers techniques: for instance the articulation of the tools on one handle is the technique that enables to design swiss army knives.

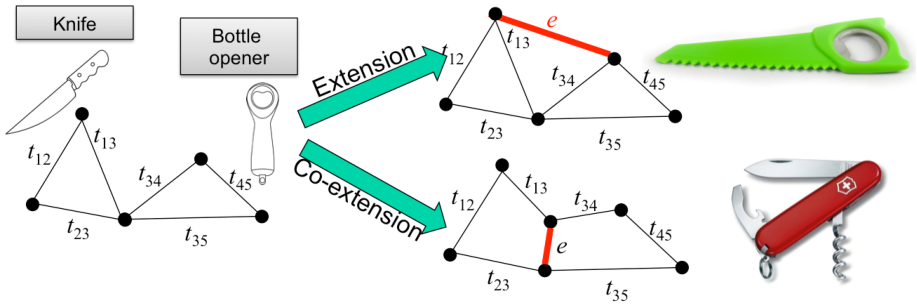


Figure 1: Illustration of the extension and co-extension on one simple case

a) Extension *preserves* the rank, creates a *new dependent* edge in a flat – hence extension creates a new working system using the new technique, this new system being a "*direct value*" of the extension. By contrast co-extension creates a *new independent* edge and *increases the rank*. The new building blocks (the new edge) is not included in a "new" working system combining building blocks that couldn't be combined before. The direct value created by the new edge is not self-evident.
b) Extension is non-pervasive: it modifies only the flats that include the newly created working system. By contrast, coextension doesn't create a new working system with the new technique and, even worse, it disturbs "old" working systems! In the figure above we see that the coextension transforms a working system $\{t_{12}, t_{13}, t_{23}\}$ into an independent set (hence no more a working system) and it is necessary to add *e* to the system to make it work again. *The new technique is now required to make work systems that worked without it before!*
c) In co-extension, the new edge connects connected components that were independent before, ie the new technique enables "bigger" working systems by aggregating smaller working systems. The new edge integrates known working systems into a bigger one. This is the critical property of coextension: *it "combines" working systems*. As such it is *pervasive*.
d) Working systems combination is "*modular*". For one "old" working system (say $\{t_{12}, t_{13}, t_{23}\}$), there are now two working systems alternatives: $\{t_{12}, t_{13}, t_{23}, e\}$ and $\{t_{12}, t_{13}, t_{34}, t_{45}, t_{53}, t_{32}\}$. *Hence coextension creates what engineering usually calls "platform" and "modularity".*

3.3 Historical illustration: the genericity of steam engine modelled with matroids.

With the illustration below we show how matroid models help to account for one famous historical case of designing a generic technology and how it confirms the most paradoxical properties of generic techniques modeled in matroids.

The story of the steam engine is often told this way: Watt designed a steam engine and progressively many applications were found for it. Yet, as shown in (Kokshagina et al. 2013a; Dickinson 1936; Thurston 1878), this story does *not* correspond to how Watt and Boulton designed a "generic" steam engine.

Actually the first generation of steam engines was adapted to mining, but not to other uses; hence there was no "steam engine" in the 1770s, but only *water pumps for mining* – and Watt was first famous, in 1770s for greatly enhancing Newcomen *"fire pumps"* with a separate condensation chamber. Actually the story of the "*generic* steam engine" begins later: in the 1780s, Boulton asked Watt to work on a new concept "a steam engine that is compatible with multiple machine tools" and it is Watt's new design for this concept, that appeared as the "steam-engine generic technology". Boulton's brief was targeting a complete recomposition of the structure of techniques of his time: if a steam engine is compatible with multiple machine tools, it becomes a core component of all future machines, as all new machine tools will be redesigned to take the best advantage of the steam engine. This "genericity"

goal corresponded for Watt to a surprisingly focused design issue: design a new *generic* technique to transmit movement from the steam engine to any other machine. This is what led Watt to the well known the "reciprocating steam engine" (see figure 2 below).

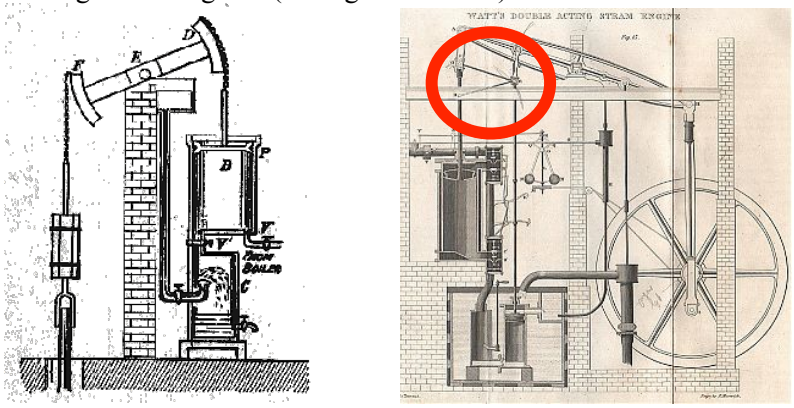


Figure 2: a) 1763 Watt steam engine with separate condensation chamber (not generic) ; b) 1784 Watt & Boulton Double acting steam engine. The parallelogram creates genericity.

In C-K/Ma the design of a "generic" steam engine relies on co-extension (see figure 3). In K: the known technique. In C, Boulton concept, and the design result: a new order of techniques based on a new technique for movement transmission. The latter appears as a "*platform*" that connects the steam engine either to mining or to workshop machine tools (in textile, iron industry, etc.). The main design effort consisted in *designing the coupling technique* (here the double acting transmission system). A pure extension logic would have led to adapt the steam engine to each of the applications.
This example illustrates many features (and paradoxes) of the design of generic techniques:

- Generic technique doesn't emerge as a complete original technology
- Despite their great systemic impact, they are discrete: the invention is neither the fire engine, nore the condensation chamber, it is only the cinematic mechanism on top of the vertical rod
- This is not an evolutionary process in which designers "discover" phenomena or "combine" randomly techniques: this is the intentional design of a pervasive technique.

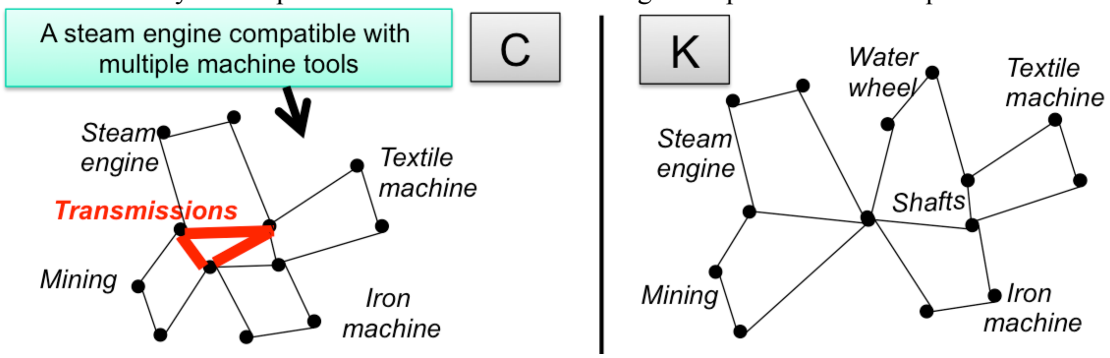


Figure 3: Designing steam engine as a generic technology

4 A COMPUTATIONAL MODEL OF THE DYNAMICS OF TECHNOLOGIES

We show now that C-K/Ma provides us with quantifiers for the dynamics of technologies (dependence, genericity, generativity), and hence helps to analyse industrial dynamics.

4.1 Characterizing features of the dynamics of technologies

Independence: The literature review helped to realize that the endogenous dynamics of technologies depends on the independence between techniques. Dependent techniques are already linked by strong relations, they can be deduced one from the other – they are deterministic. On the other hand independent techniques are *not yet* related so they are "holes" in the knowledge base.
With C-K/Ma we can *quantify independence and dependence*: the rank of the matroid M, r(M), gives the level of independence; the co-rank r* gives the level of dependence (r* is the rank of the dual M*, ie the independence between working systems.). Both are linked by the equation: $r(M) + r(M^*) = |M|$ where $|M|$ is the number of edges, ie the number of technological building blocks.

Generativity: Generativity is a critical feature of any design theory (Hatchuel et al. 2011). It is usually hardly quantified. In C-K/Ma generativity can be seen as the number of new edges (techniques) that can be created on a given matroid, either by extension or by co-extension. Hence we *quantify the generativity by the number of possibilities of single edge extension* $g_{ext}(M)$ *and of single edge coextension* $g_{co\text{-}ext}(M)$ *(see figure 4 below):* $\mathbf{g_{ext}(M) = r(r-1)/2-r^*+p}$ and $\mathbf{g_{co\text{-}ext}(M)= r^*(r^*-1)/2-r+p^*}$ (where p and p* are the number of loops or parallel edges in M and M*). $g_{ext}(M)$ is rather the maximum number of basic new working systems while $g_{co\text{-}ext}(M)$ *estimates here the maximum quantity of "generic" techniques that can be proposed on a set of given techniques*.

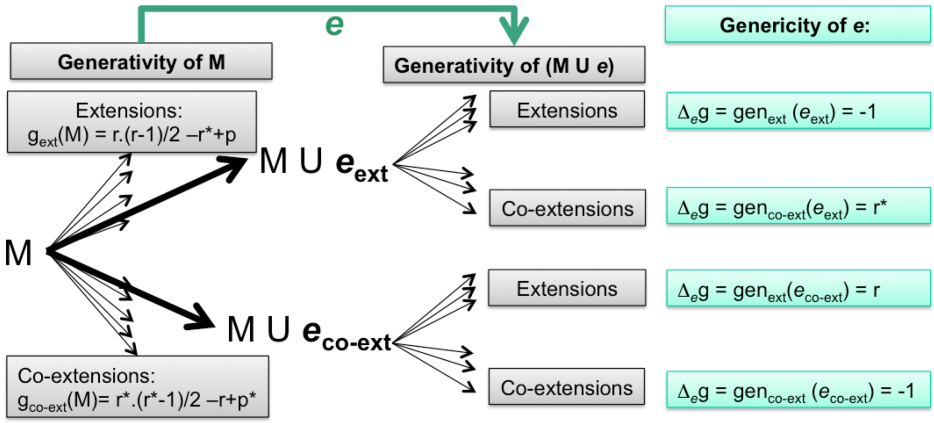


Figure 4: Generativity of a matroid, genericity of a new edge added to this matroid

Genericity: With genericity we mean the number of applications *derived from one design* (Kokshagina et al. 2013a). *Genericity characterizes the systemic impact of one particular new technique*. We quantify genericity by the number of new circuits that can be created after the design of *e*, with one additional extension or coextension. We call N=M ∪ e the matroid created by e added to M. If *e* results from an extension, then r(N) = r(M) and r*(N)= r*(M)+1. Hence $\mathbf{gen_{ext}\ (e_{ext}) = -1}$. (Similarly $gen_{co\text{-}ext}\ (e_{co\text{-}ext}) = -1$). If *e* results from a co-extension, then r(N) = r(M) + 1 and r*(N)=r*(M). Hence $\mathbf{gen_{ext}(e_{co\text{-}ext}) = r}$ (conversely: $gen_{co\text{-}ext}(e_{ext}) = r^*$). (see figure 4)

4.2 Illustration: designing generic technologies in the semiconductor industry

We use these quantifiers to analyze the design of generic technique in semiconductor industry (Kokshagina et al. 2013b). The concept consists in combining three unrelated technological families:

- Family 1: the computing technological building blocks (CMOS transistor)
- Family 2: the Radio-Frequence sensors, to receive and digitalize radio frequency signals
- Family 3: the "back-end" system, in charge of routing and processing signals

By combination one expects a system that enables circuits using building blocks in two or the three families. The notion of "combination" is usually quite fuzzy and hides the design issues. We apply the matroid model. Combining extensions and co-extensions, one can propose four different solutions (see figure below) and *compare the genericity created by each design alternative*.

- The "pure extension" appears as an effort *to design "micro-combinations" of building blocks*; it brings multiple working systems –hence a direct value-, still it doesn't enable to combine *all* previously known building blocks. The genericity is negative: the new design has decreased the generativity potential of the matroid.
- The "pure co-extension" strategy creates a working system that uses all previously known techniques; it is modular. It creates less working systems than in "pure extension" case. The genericity is very high: the new design has increased the generativity potential of the matroid.
- Hybrid case 1 exhibits a new connected sub-component that connects each of the modular ones. We see here a "*platform*". Genericity is high.
- Hybrid case 2 enables to get the expected performance (connections between the connected components) *by designing only two edges*, whereas all the other solutions design three edges.

The design strategy adopted at STMicroelectronics actually follows the latter process: mixing CMOS and bipolar into a new connected component called bi-CMOS (hence an extension) and the redesign of the back-end to connect it to the new bi-CMOS (co-extension).

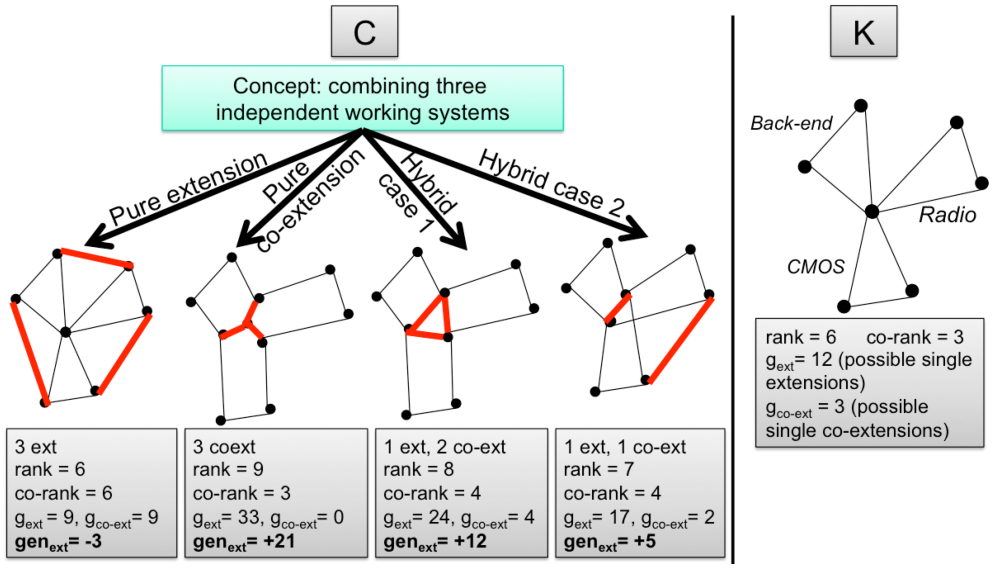


Figure 5: Combining technologies – the case of B9MW in semiconductors

This example shows how C-K/Ma and the quantifiers on generativity and genericity enrich our understanding of the strategies to design techniques with systemic impact.

4.3 Characterizing the dynamics of techniques

C-K/Ma also uncovers general rules on the dynamics of techniques. First negative results:

1- **Lego like structure of techniques implies exogenous dynamics.** Let's analyse the intuitive model of "combination" (used in (Arthur 2009)). In this model, each building block can be combined with another one to create a working system. This "legolike" structure corresponds to one specific matroid, namely U(1,n). In U(p,n) the independent sets are the sets containing less than p elements taken among n. We have $g_{ext}(U(1,n)) = 0$: extension is impossible in U(1,n). The "pure combination" approach prevents the design of new working systems! (actually because all working systems are known in the legolike structure). We also have: $g_{co-ext}(U(1,n)) = (n-1).(n-2)/2-1$. Hence U(1,n) is very generative in *coextension*. However, these coextensions are impossible in "lego-like" approaches *because the co-extended matroid is no more Lego-like* (the U(1,n) family is not stable by coextension). *Hence a Legolike structure can not account for the endogenous dynamics of technological systems.* It explains why intuitive "combination" models tend to rely on exogenous dynamics.

2- Purely independent techniques implies exogenous dynamics. Let's analyse another "intuitive" model of systems. We consider now the n techniques in a working system that follows the first axiom of Suh. This is a working system, hence the n techniques form a circuit. Now the first axiom applies "inside the working system", which means that the dependence linking all techniques to form a working system is neglected and any subset of n-1 techniques follows the first axiom. They are independent. Hence the set of techniques enabling a Suh working system might be represented by a U(n-1, n) matroid. In such a matroid, no co-extension is possible. Ie *Suh's first axiom prevents the design of generic techniques. Hence a "perfect" system engineering will finally prevents the emergence of a generic technique*! Conversely by co-extension U(n-1, n) is not stable so that the newly created structure is no more following Suh Axiom. Hence it is not possible to account for endogenous dynamics if one considers that the techniques follow the Suh axiom of independence.

3- A theorem for continuous endogenous dynamics: the necessary combination of extension and coextension. The two simplified representations of technical systems – pure combination or pure independence- actually correspond to very particular matroids and we showed that they can not account for the dynamics of techniques. More generally: *since the extension-genericity of extension is negative (decreasing generativity) and the coextension-genericity of co-extension as well, extensions or coextensions alone lead to deadlocked systems.* A direct consequence of the negative genericity is a general theorem for the continuous endogenous dynamics of technique: *the only way to get an unlocked endogenous dynamics consists in combining extension and coextension – ie the combination of the design of working system and the design of generic techniques.*

5 MAIN FINDINGS AND CONCLUSION

We built a model of the design of knowledge structure by combining design theory (C-K theory) and a model of independence in knowledge structures (matroid theory). Findings are:

1) F1: C-K/Ma offers a computational model for designing a technique with systemic impact. C-K/Ma characterizes main operations (extension, co-extension) and their effect (rank, co-rank, generativity, genericity). This model helps to overcome misleading intuitions. We show that lego-like "combinative" model as well as Suh independence prevent endogenous dynamics.

2) F2: C-K/Ma accounts for phenomena associated to generic technology design:

a) a generic technique does not seem to add functional value but is finally in every machines as the key technique to combine previously independent working systems;
b) a generic technique *couples and decouples*, it creates a "modular" relations between working systems – it reorganizes technical systems in a flexible way.
c) A generic technique opens new opportunities: new techniques can be added to the newly coupled working systems.

3) F3: C-K/Ma offers an efficient guide for the design of technologies with systemic impact, based on generativity and genericity criteria. It uncovers the variety of design strategies for pervasive techniques.

Finally, relying on advanced models of design theory and matroid, this paper strengthens the study of the endogenous dynamics of technical systems and opens new ways. Beyond the phenomena and the strategy, the model might also be useful to study economics and organizational issues. In particular the logic of "indirect" value in the design of generic technology actually raises interesting institutional issues: are companies – more interested for direct value- able to design generic technologies with high indirect value? Who could be the new actors in charge of these designs? To study the design of generic techniques we integrated in the model the logics of cohesion and interdependences between techniques – this might now lead us to shed new light on the logics of cohesion and interdependences in economics and society, in particular among designers and in their societies, like the Design Society.

REFERENCES

Arthur WB (2009) *The Nature of Technology. What it is and how it evolves.* . Free Press, New York

Baldwin CY, Clark KB (2000) *Design Rules, volume 1: The power of modularity*. The MIT Press, Cambridge, MA, USA

Braha D, Reich Y (2003) Topologial structures for modelling engineering design processes. *Research in Engineering Design* 14 (4):185-199.

Bresnahan TF, Trajtenberg M (1995) General Purpose Technologies: Engines of Growth? *Journal of Econometrics* 65 (1):83-108.

Dickinson HW (1936) *Matthew Boulton*. Réédition de 1999 edn. TEE Publishing, Warwickshire, England

Dosi G, Freeman C, Nelson R, Silverberg G, Soete L (eds) (1988) *Technical Change and Economic Theory*. London

Gawer A (ed) (2009) *Platforms, Markets and Innovation*. Edward Elgar, Cheltenham, UK and Northampton, MA

Hatchuel A, Le Masson P, Reich Y, Weil B (2011) A systematic approach of design theories using generativeness and robustness. In: *International Conference on Engineering Design, ICED'11*, Copenhagen, Technical University of Denmark, 2011. p 12

Hatchuel A, Weil B, Le Masson P (2013) Towards an ontology of design: lessons from C-K Design theory and Forcing. *Research in Engineering Design* 24 (2):147-163.

Kokshagina O, Le Masson P, Weil B (2013a) How design theories enable the design of generic technologies: notion of generic concepts and Genericity building operators Paper presented at the *International Conference on Engineering Design, ICED'13*, Séoul, Korea,

Kokshagina O, Le Masson P, Weil B, Cogez P (2013b) Platform emergence in double unknown (technology, markets): common unknown strategy. In: Çetindamar D, Daim T, Başoğlu N, Beyhan B (eds) *Strategic planning decisions in the high tech industry*. Springer, London, pp 90-120

Le Masson P, Weil B (2013) Design theories as languages for the unknown: insights from the German roots of systematic design (1840-1960). *Research in Engineering Design* 24 (2):105-126.

Le Masson P, Weil B, Kokshagina O (2015) A new perspective for risk management: a study of the design of generic technology with a matroid model in C-K theory. In: Taura T (ed) *Principia Designae – Pre-Design, Design, and Post-Design - Social Motive for the Highly Advanced Technological Society*. Springer, Tokyo, pp 199-219

Neel DL, Neudauer NA (2009) Matroids you have known. *Mathematics magazine* 82 (1):26-41.

Oxley J (2011) *Matroid Theory*. Oxford Graduate Texts in Mathematics, 2nd edition edn. Oxford University Press,

Reich Y (1995) A Critical Review of General Design Theory. *Research in Engineering Design* 7:1-18.

Saviotti PP, Metcalfe JS (1991) *Evolutionary Theories of Economic and Technological Change*. Harwood Academic Publishers, Newark, N.J.

Shai O, Reich Y (2004a) Infused Design: I Theory. *Research in Engineering Design* 15 (2):93-107.

Shai O, Reich Y (2004b) Infused Design: II Practice. *Research in Engineering Design* 15 (2):108-121.

Shai O, Reich Y, Hatchuel A, Subrahmanian E (2013) Creativity and scientific discovery with infused design and its analysis with C-K theory. *Research in Engineering Design* 24 (2):201-214.

Suh NP (2001) *Axiomatic Design: advances and applications*. Oxford University Press, Oxford

Thurston RH (1878) *A History of the Growth of the Steam Engine*. Appleton, New York

Ulrich KT, Eppinger SD (2008) *Product Design and Development*. 4th edn. Mc Graw Hill,

Whitney H (1935) On the Abstract Properties of Linear Dependance. *American Journal of Mathematics* 57 (3):509-533.

Yoshikawa H (1981) General Design Theory and a CAD System. In: Sata T, Warman E (eds) *Man-Machine Communication in CAD/CAM, proceedings of the IFIP WG5.2-5.3 Working Conference 1980 (Tokyo)*. Amsterdam, North-Holland, pp 35-57